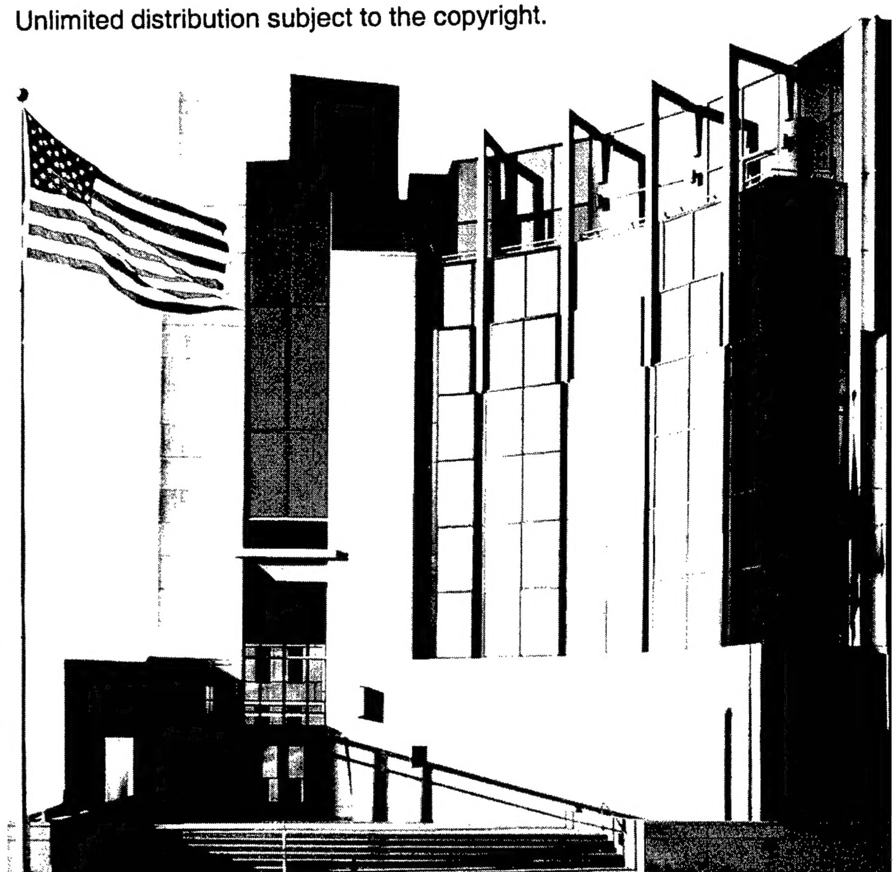
Carnegie Mellon
Software Engineering Institute

Options Analysis for Reengineering (OAR): A Method for Mining Legacy Assets

John Bergey
Liam O'Brien
Dennis Smith

June 2001

Product Line Practice Initiative



Technical Note
CMU/SEI-2001-TN-013

Options Analysis for Reengineering (OAR): A Method for Mining Legacy Assets

John Bergey
Liam O'Brien
Dennis Smith

June 2001

Product Line Practice Initiative

Technical Note
CMU/SEI-2001-TN-013

For information about purchasing paper copies of SEI reports, please visit the publications portion of our Web site (http://www.sei.cmu.edu/publications/pubweb.html).

Contents

List of Figures

List of Tables

Abstract

Options Analysis for Reengineering (OAR) is a systematic, architecture-centric, decision-making method for mining existing components for a product line or new software architecture. OAR's five activities identify potential components, estimate the mining cost, and evaluate the effort required to reuse legacy components. OAR reveals implicit stakeholder assumptions, constraints, and other major drivers that affect component mining, thereby giving managers insight into this complex task.

1 Introduction

Options Analysis for Reengineering (OAR) is a systematic, architecture-centric, decision-making method for identifying and mining software components within large, complex software systems. Mining involves rehabilitating parts of an old system for reuse. OAR identifies potentially relevant architectural components and analyzes the changes required to use them in a software product line or new software architecture. In essence, OAR provides a set of mining options along with estimates of the cost, effort, and risks associated with those options.

1.1 Need for OAR

Many organizations have undertaken efforts to update their software intensive systems and to establish more efficient ways of developing software. One emerging trend has been the implementation of software product lines to realize economies of scale and greater efficiencies in software development [Clements 01].

Since most organizations have a substantial legacy base of existing software assets, few product line development or migration efforts start from "green fields." However until now, there has been no systematic way to identify components for reuse and to understand the types of changes required to insert legacy system components into a software product line or a new software architecture [Muller 00]. In most cases, the options have been to either undergo the costly and high-risk process of reengineering an entire system or to simply build the required components or systems from scratch.

Component mining has often been discussed as an alternative, but requires understanding what types of components are worth extracting and how to extract them. The following issues add to the challenge:

- Existing components are often poorly structured and poorly documented.
- Existing components differ in levels of granularity.
- There is no clear guidance on how to salvage components.

OAR provides a systematic approach to addressing these issues and to making the decisions required to cost-effectively and efficiently mine legacy system components.

1.2 Structure of This Technical Note

This technical note provides a brief overview of OAR. Section 2 outlines the activities of OAR and provides a brief summary of the purpose and tasks of each activity. Section 3

describes how each activity is structured according to an EITVOX (Entry Criteria, Inputs, Tasks, Validation, Outputs, Exit Criteria) format. It describes one activity "Establish Mining Context" to demonstrate how this format is followed. Section 4 provides an example of how OAR has been used. Section 5 provides a summary of OAR, its status and potential. Bergey et al., provide additional details on OAR in their OAR tutorial [Bergey 01].

2 The OAR Method

The OAR method consists of five major activities with scalable tasks. These tasks are depicted in Figure 1 and summarized in the following subsections:

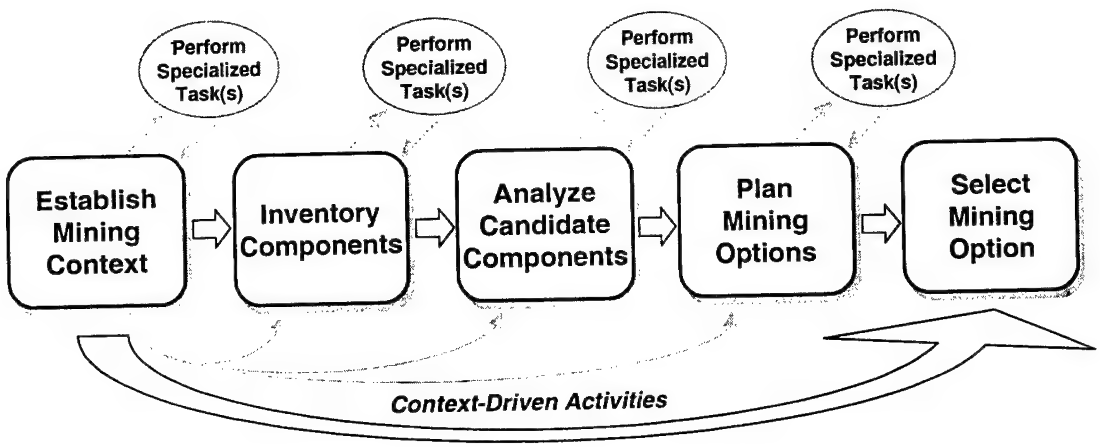


Figure 1: Overview of OAR Activities

2.1 Establish Mining Context

It is important for the OAR team to understand the context for the mining effort. As a result, the first activity of OAR involves interviewing stakeholders and studying the organization's product line or new system requirements, legacy base, and expectations for mining legacy components. This effort establishes a baseline set of goals, expectations, and component needs. It also uncovers the program and technical drivers for making decisions.

2.2 Inventory Components

Next, the OAR team identifies the legacy system components that potentially can be mined for use in a product line or new software architecture.

During this activity, team members identify product line component needs and evaluate the legacy components based on these criteria. Those that don't meet the criteria are screened out. This activity results in an inventory of candidate legacy components.

The Inventory Components activity has six tasks:

1 Identify Characteristics of Component Needs:
Determine required characteristics of potential legacy components.

2 Identify Components Satisfying Characteristics:
Create a Component Table of the legacy components with details of these characteristics.

Screen components that do not satisfy the required characteristics.

3 Match Components to Needs:
Match legacy components to product line component needs.

4 Inventory Candidate Components:
Update the Component Table with more details about those candidate components chosen.

5 Elicit Mining Issues:
Review any mining issues and concerns that were identified during the activity.

6 Review OAR Schedule:
Update the OAR Schedule, if necessary.

2.3 Analyze Candidate Components

Next, team members analyze the candidate set of legacy components for the types of changes that are required to mine. This activity has six tasks:

1. Select Desirable Components:

Determine desirability criteria for each legacy component.

Screen out those that do not satisfy these criteria.

1. Identify As Is & Black-Box Components:

Determine those components that can be wrapped or used "as is."

1. Identify White-Box Components:

Determine those components that will need to be modified.

1. Determine Required Changes:

Determine the types of changes that each component will need, the cost and effort involved, the level of difficulty and risk, and the comparative cost and effort of developing components

from scratch.

1. Elicit Mining Issues:

Review mining issues and concerns that were identified during the activity.

1. Review OAR Schedule:

Update the OAR Schedule, if necessary.

2.4 Plan Mining Options

Given the set of analyzed candidate components, the team next develops alternatives to mining based on schedule, cost, effort, risk, and resource considerations. The OAR team also screens candidate components one more time and analyzes the impact of different component aggregations.

The Plan Mining Options activity has seven tasks:

1. Select Favorable Components:

Develop criteria, such as cost or effort levels required.

Screen components that do not satisfy the criteria.

2. Perform Component Tradeoffs:

Identify one component (or combination) per product line component need.

3. Form Component Options:

Develop criteria for aggregating, and aggregate components.

4. Determine Comparative Cost and Effort:

Compare the cost of each aggregation with the cost of developing it from scratch.

5. Analyze Difficulty and Risk:

Determine the level of difficulty and the risks involved for each aggregation.

6. Elicit Mining Issues:

Review mining issues and concerns identified during the activity.

7. Update OAR Schedule:

Update the OAR Schedule, if necessary.

2.5 Select Mining Option

Finally, team members select the best mining option or combination of options by balancing program and technical considerations. After evaluating each mining option, they prepare a summary report that presents and justifies their selections.

The Select Mining Option activity has five tasks:

1 Choose Best Option:

Determine the drivers for selecting from among the options

Select an option or combination of them.

2 Verify Option:

Record all the details about each of the options chosen.

3 Identify Component Needs Satisfied:

Complete the final list of component needs satisfied and not satisfied through the options selected.

4 Present Findings:

Prepare findings presentation that provides details about the options selected.

5 Produce Summary:

Produce a final report detailing the options chosen and the rationales for their selection.

2.6 Specialized Tasks

Each activity also has a set of potential specialized tasks to address circumstances that may otherwise preclude completing the activity. These tasks may apply under the following conditions:

- The exit criteria for the prescribed activity have not been satisfied.
- More work is required than can reasonably be accomplished in the activity wrap-up tasks.
- Additional data is needed to complete a particular task or to address special customer needs.

- There is a need to supplement standard OAR tasks to enhance decision-making and reduce risk.

Section 3 includes examples of specialized tasks.

3 Structure of Activities

Each activity is composed of tasks and sub-tasks designed to answer a set of activity-specific questions. These questions will define the activity and will also serve as a checklist to be included in the exit criteria for each activity.

The activities are structured according to an EITVOX (Entry Criteria, Inputs, Tasks, Validation, Outputs, Exit Criteria) format as follows:

Table 1: EITVOX Structure of OAR Activities

Prescribed OAR Activity

E	**Entry Criteria**	Criteria for beginning activity
I	**Inputs**	Artifacts required during execution of activity
T	**Tasks**	Tasks to carry out the prescribed activity
V	**Validation**	Data collected to validate the activity
O	**Outputs**	Artifacts produced during execution of activity
X	**Exit Criteria**	Criteria defining completion of activity

At the end of each activity, examples are given of the specialized tasks that may need to be performed before continuing.

3.1 Overview of Sample Activity: Establish Mining Context

The following subsections show how the first activity, Establish Mining Context, is structured according to the EITVOX format. Each of the other activities follows this format. Bergey et al., provide this level of detail [Bergey 01].

The purpose of Establish Mining Context is to understand the organization's product line or new single system needs, legacy base, and expectations for mining legacy components.

3.1.1 Underlying Questions

The tasks of Establish Mining Context were developed to address a set of underlying questions that include

- What are management's expectations for the mining effort?
- What are the requirements (e.g., quality attributes) and primary drivers (e.g., priorities and constraints) for mining components?
- What explicit (and implicit) constraints, ground rules, and assumptions apply?
- What specific product line component needs will be addressed?
- What legacy systems are relevant and what documentation is available?
- What documents are available describing legacy components (e.g., their functionality, granularity, and interfaces)?
- What is the level of preparedness of the organization in terms of experience, skills, available resources, and OAR timeframe?

The complete set of underlying questions is revisited during validation to make sure they were answered satisfactorily.

3.1.2 Entry Criteria

The following entry criteria are examined to determine if there is sufficient data and expertise available to perform the activity successfully. The entry criteria for this activity are

- The organization has decided to move to a product line and wants to quickly assess the viability of mining components.
- The product line architecture has been defined and component needs have been identified.
- A set of mining goals and objectives has been established.
- One to three legacy systems have been identified as the source of components for mining effort.
- A "Mining Team" has been chartered and is available for the duration of OAR activities.

If the entry criteria are not satisfied, a specialized task may need to be developed.

3.1.3 Input Artifacts

Each activity relies on a set of support artifacts. The following artifacts are required for Establish Mining Context:

- goals and objectives for the mining effort
- product line requirements and component needs
- documentation describing product line scope, architecture, and component specifications

- overview of legacy system and component documentation
- experience reports of other mining/reengineering efforts
- typical OAR Scheduling Profile[1]

3.1.4 Task Structure

The Establish Mining Context activity is divided into 10 tasks. These tasks are executed in order. Several of the tasks are further divided into subtasks. To help OAR team members perform tasks and sub-tasks, the SEI developed execution and data templates. Execution templates outline the steps, rationale, and assumptions for the task or sub-task. Data templates provide example topics and often serve as a starting point for eliciting information from a customer.

The tasks accomplish the following:

1 Review Goals and Objectives:

Determine stakeholder goals and objectives for the mining effort.

2 Review Product Line/New System Requirements:

Identify product line or new system requirements.

3 Review and Select Component Needs:

Identify component needs that will be addressed by mining.

4 Review Legacy Systems and Documentation:

Review the legacy systems and component documentation available.

5 Determine Mining Drivers:

Identify programmatic and technical drivers.

6 Validate Goals and Objectives:

Determine compatibility of mining drivers with goals and objectives.

7 Identify Candidate Components:

Determine criteria for high-value legacy components.

Select a set of candidate components.

8 Elicit Mining Issues:

[1] The OAR Scheduling Profile is a template provided by the SEI.

Review any mining issues and concerns that were identified during the activity.

9 Evaluate Preparedness:

Determine the organization's level of preparedness.

10 Review OAR Schedule:

Update the OAR Schedule, if necessary.

Some tasks have sub-tasks. Task 4, for example, is subdivided into:

1 Review Legacy Systems.

2 Review Component Documentation.

3.1.5 Example of Execution and Data Templates

The execution template, illustrated in Table 2, specifies how to perform Review Component Documentation (sub-task 4.2).

The data template "EMC-4.2-DT" suggests that the following type of data be collected:

- list of legacy components
- component documentation (i.e., functionality and interface descriptions)
- source code for legacy components
- maintenance history for legacy components (e.g., cost, modification, timeframe, resource utilization)

Table 2: Task Execution Template: EMC-4.2-DT

Task ID:	EMC Task 4.2
Name:	Review Component Documentation.
Description:	As a precursor to mining, review what type of documentation is available at the component level.
Steps:	1. A representative of the customer organization conducts a walkthrough describing the type and level of documentation available on the legacy system(s) using data template EMC-4.2-DT as a guide: Key topics that should be addressed include: What is the level of granularity for legacy components?

	How does this level of granularity compare to product line needs? What component documentation is typically available at that level of granularity? 2. Review and consider the information that is presented and ask questions to clarify and refine the type and level of documentation that is available to support the analyses that the mining team will be performing. 3. Identify any documentation deficiencies and their potential impact.
Rationale:	There is a need for the two parties to establish beforehand what documentation is available to support inventory and analysis. If documentation is lacking, determine the potential impact on the mining effort.
Assumptions:	Refer to the assumptions under Establish Mining Context Task 7.3 for specific courses of action to compensate for documentation deficiencies. If particular subsystems are poorly documented, the mining team may want to eliminate the components of those subsystems from further consideration. This is in keeping with the underlying concept of the OAR "bridging" scenario; i.e., quickly identify those systems and subsystems that have the most potential for mining, and select high value components from them.

3.1.6 Validation

After the tasks are completed, the underlying questions for the activity are reviewed. The following are some of the validation criteria for Establish Mining Context:

- Expectations are outlined and understood.
- All relevant legacy systems are identified.
- A set of 10 to 20 product line component needs have been selected with high potential to be satisfied through mining.
- Drivers and priorities are identified and understood.
- A set of 15 to 30 legacy components have been selected as mining candidates.
- The level of mining preparedness has been evaluated.

3.1.7 Output Artifacts

Establish Mining Context produces the following output artifacts:

- a set of relevant legacy systems and documentation
- a set of selected product line component needs
- major program and technical drivers

- a set of selected legacy components and associated component documentation
- preparedness evaluation
- list of mining issues and concerns
- revised OAR Scheduling Profile

3.1.8 Exit Criteria

Before completing the Establish Mining Context, the following exit criteria need to be satisfied:

- The mining team has identified a reasonable set of product line needs and candidate components.
- The organization's expectations are consistent with its level of preparedness.
- The OAR Scheduling Profile has been reviewed and any needed changes have been accepted.
- All output artifacts of this activity have been produced.
- All underlying questions for this activity have been answered satisfactorily.

3.1.9 Specialized Task Examples

If the exit criteria are not satisfied, a specialized task may be appropriate. Examples of specialized tasks for the Establish Mining Context activity follow:

- Further identifying mining effort drivers and resolving programmatic and technical issues.
- Further defining product line requirements and component needs in terms of functionality and interfaces.
- Isolating and identifying the functionality and interfaces of legacy components.
- Developing the required level of documentation for legacy components.

4 An Example of an Application of OAR

OAR was used to explore component mining for a Satellite Tracking Agency. This agency supports efforts to develop, acquire, and deploy satellite tracking systems. The agency had an urgent need to mine components from an existing Satellite Tracking System (STS) for the architecture in their Improved Satellite System (ISS). ISS will integrate space assets with a consolidated ground segment. ISS requires some of the capabilities of STS. Because STS already has a set of reliable and validated models, the agency decided to mine and reuse selected components in ISS.

Members of the technical staff at the SEI and agency representatives formed an OAR team to identify relevant components and to estimate the cost and effort required to rehabilitate them for the ISS. The team first established a set of drivers for making decisions on candidate components. These drivers included flexibility of interfaces, satisfaction of real-time constraints, portability and interoperability, and high level of code quality (i.e., high cohesion, low coupling). After examining the STS system, the team selected six legacy system components as mining candidates: two simulation gateways, a message gateway, two truth model components, and a theater event sub-system. The Component Table that appears in Appendix A describes each component.

Three potential wrapping components were identified (the two simulation gateways and the message gateway). These components would not require modifying their internal programs. The other three components (the two truth model components and the theater event sub-system) were identified as potential white box components that would require changes.

Next, the team estimated the basic types of changes required to rehabilitate these components. The OAR team also estimated the effort and cost to develop candidate components from scratch and compared these figures to the cost of mining them. However, since the interfaces for the new system had not yet been fully defined, the cost estimates were preliminary. Once the ISS interface specifications are complete, the cost estimates will need to be reviewed.

Three options for aggregation were developed. The gateway and message components were aggregated as one option. However, since these components all depend on the interfaces that are still undefined, this aggregation may change. The other two components represent coarse-grained groupings of self-contained functionality. They were each treated as separate options. The team then calculated the comparative cost and effort of each aggregation based on the information captured in the Component Table. The team also determined the aggregated level of difficulty and risk for each option. The Options Table in Appendix B presents this data.

5 Conclusion

OAR is a systematic, architecture-centric, decision-making method for mining existing components. It identifies software product line or new system architecture component needs that can be satisfied through mining, as well as those that cannot be satisfied. Candidate components are screened based on criteria that are relevant for the organization. OAR provides management visibility into this highly complex analysis activity. It also provides insights into implicit stakeholder assumptions, constraints, and other major drivers that impact the mining of components. OAR determines the feasibility of mining components at an early stage of a reengineering project. OAR provides insight into technical issues as well as cost and effort estimates and helps to address the problem that many reengineering projects fail due to lack of early insight into the magnitude of an effort.

As a first step OAR establishes the component needs, identifies candidate legacy components and their related documentation, and determines other aspects that define the mining context. Next it inventories the legacy components that can potentially fill identified product line needs, and identifies their characteristics, types of changes needed, and estimated cost and effort. OAR reflects the elicited needs, priorities and concerns of the organization. Its final outputs are a mining option or combination of options that an organization may pursue and a list of product line component needs that can and cannot be satisfied through mining.

OAR has been applied successfully at the Satellite Tracking Agency, and lessons learned from this effort led to the current Version 2. We are planning to apply OAR on other projects and continue to refine and improve the method. We are currently seeking organizations interested in applying the method. In the near future, we will develop a handbook to accompany the method. Eventually, we plan to transition the method to other organizations.

References

[Bergey 01] Bergey, J.; O'Brien, L. & Smith, D. *Options Analysis for Reengineering (OAR).* Tutorial presented at ICSE 2001. Toronto, Ontario: May 2001.

[Clements 01] Clements, P. & Northrop, L. *Software Product Lines: Practice and Patterns.* Boston, MA: Addison Wesley Longman, Inc., 2001.

[Muller 00] Muller, H.; Jahnke, J.; Smith, D.; Storey, M.; Tilley, S. & Wong, K. "Reverse Engineering: A Roadmap." 47-60. *The Future of Software Engineering*. New York, NY: ACM, 2000.

Appendix A. Component Table

Part 1 of 3		Legacy System Software					
		Characteristics				Characteristics	
COMPONENT NEEDS	Legacy System Software Components	Program Language	Directory Name	Number of Modules	Compilable on SGI Machine	Size[1] Lines of Code (LOC)	Complexity
1. Simulation Gateways	Sim1 Gateway	C	cag	17	Yes	2,297	Low
	Sim2 Gateway	C	dwg	57	Yes	4,877	Low
2. Message Gateways	C-Source	C	csg	45	Yes	4,186	High
3. Truth Model	Objects	Fortran	ojo	65	Yes	9,202	Med
	Event	Fortran	tho	39	Yes	4,412	Med
4. ETS & Message Generation	Event Track	C	tsd	89	Yes	5,417	Med / High
5. Man Machine Interface	TBD					30,391 ◄Total►	
6. Executive	TBD						

●●●

PART 1 of 3

Footnotes:
1. Size estimates are based on actual count of source lines for each module.

Part 2 of 3

Legacy System Software

Legacy System Software Components	Characteristics: Coupling	Characteristics: Cohesiveness	Characteristics: Black Box / White Box Suitability	Characteristics: Level of Changes	Characteristics: Level of Granularity	Support Software Required	Level of Difficulty[3]	Level of Risk
Sim1 Gateway	Low	High	BB	None	OK	S&D Files[2]	1	Low
Sim2 Gateway	Low	High	BB	None	OK	S&D Files	1	Low
C-Source	Low	High	BB	None	OK	S&D Files	1	Low
Objects	High	High	WB (minor)	Minor (10%)	Adjust	S&D Files	2	High[4]
Event	High	High	WB (minor)	Minor (10%)	Adjust	S&D Files	2	High[4]
Event Track	Low	Low	WB (major)	Major (50%)	Adjust	S&D Files	4	Low[5]
TBD								
TBD								

PART 2 of 3

Footnotes:

2. S&D Files are Scripts and Data Files.
3. Level of difficulty is identified on a scale of 1 (Minimal) to 5 (Very High).
4. These are outside of the control of the organization.
5. These are under the control of the organization.

Part 3 of 3

Legacy System Software

Software Components	Mining Effort[6] (mm)	Mining Cost[7]	New Development Effort (mm)	New Development Cost[8]	Comparative Cost[7] of Mining	Comparative Effort[6] of Mining
Sim1 Gateway	0.1	$1,000	7.7	$76,567	1%	1%
Sim2 Gateway	0.1	$1,000	16.3	$162,567	1%	1%
C-Source	0.1	$1,000	14.0	$139,533	1%	1%
Objects	2.2	$22,000	30.7	$306,733	7%	7%
Event	1.3	$13,000	14.7	$147,067	9%	9%
Event Track	8	$80,000	18.1	$180,567	44%	44%
TBD	12	$118,000	101	$1,013,033	12%	12%
	← TOTALs →					
TBD						

PART 3 of 3

Footnotes:

6. The labor for converting the scripts and data files that are part of the support software is: 6 man-months. This estimate is not included in the total effort.

7. The added cost of converting the scripts and data files that are part of the support software is: $60,000 This cost, which is based on the effort estimate above, is not included in the total mining cost.

8. The new development cost estimates are based on an average cost of $33.33 per line of code (LOC).
This figure is based on the following entries:
$10,000 Burdened labor rate per man-month (mm)
300 Estimated productivity rate (LOC per man-month)

Appendix B. Options Table

Option No.	Legacy System Software Components	Support Software Required	Level of Risk	Level of Difficulty	Mining Effort[1] (mm)	Mining Cost[1]	New Development Effort (mm)	New Development Cost	Comparative Cost of Mining	Comparative Effort of Mining
1	Event Track	S&D Files	Low	4	8	$80,000	18.1	$180,567	44%	44%
	Option Summation		**Low**	**4**	**8**	**$80,000**	**18.1**	**$180,567**	**44%**	**44%**
2	Objects	S&D Files	High	2	2.2	$22,000	30.7	$306,733	7%	7%
	Event	S&D Files	High	2	1.3	$13,000	14.7	$147,067	9%	9%
	Option Summation		**High**	**2**	**3.5**	**$35,000**	**45.4**	**$453,800**	**8%**	**8%**
3	Sim1 Gateway	S&D Files	Low	1	0.1	$1,000	7.7	$76,567	1%	1%
	Sim2 Gateway	S&D Files	Low	1	0.1	$1,000	16.3	$162,567	1%	1%
	C-Source	S&D Files	Low	1	0.1	$1,000	14	$139,533	1%	1%
	Option Summation		**Low**	**1**	**0.3**	**$3,000**	**38**	**$378,667**	**1%**	**1%**

[1] Note: Mining Effort and Cost do not include effort and cost to convert scripts and data files that are part of the support software

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY
(Leave Blank)

2. REPORT DATE
June 2001

3. REPORT TYPE AND DATES COVERED
Final

4. TITLE AND SUBTITLE
Options Analysis for Reengineering (OAR): A Method for Mining Legacy Assets

5. FUNDING NUMBERS
F19628-00-C-0003

6. AUTHOR(S)
John Bergey, Liam O'Brien, Dennis Smith

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

8. PERFORMING ORGANIZATION REPORT NUMBER
CMU/SEI-2001-TN-013

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
HQ ESC/XPK
5 Eglin Street
Hanscom AFB, MA 01731-2116

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12A DISTRIBUTION/AVAILABILITY STATEMENT
Unclassified/Unlimited, DTIC, NTIS

12B DISTRIBUTION CODE

13. ABSTRACT (MAXIMUM 200 WORDS)
Options Analysis for Reengineering (OAR) is a systematic, architecture-centric means for mining existing components for use in a product line or new software architecture. OAR's five activities identify potential components, estimate the mining cost, and evaluate the effort required to reuse legacy components in a new software architecture. OAR reveals implicit stakeholder assumptions, constraints, and other major drivers that affect the component mining, thereby giving managers insight into this complex task.

14. SUBJECT TERMS
software architecture, reengineering, mining, software components, reuse, software product lines, evaluation

15. NUMBER OF PAGES
31

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT
Unclassified

18. SECURITY CLASSIFICATION OF THIS PAGE
Unclassified

19. SECURITY CLASSIFICATION OF ABSTRACT
Unclassified

20. LIMITATION OF ABSTRACT
UL

NSN 7540-01-280-5500

Standard Form 298 (Rev 2-89) Prescribed by ANSI Std. Z39-18 298-102